

KNOBBE, MARTENS, OLSON & BEAR, LLP  
620 NEWPORT CENTER DR 16TH FLOOR NEWPORT BEACH, CA 92660  
(949) 760-0404 FAX (949) 760-9502

## KNOBBE, MARTENS, OLSON &amp; BEAR

A LIMITED LIABILITY PARTNERSHIP INCLUDING  
PROFESSIONAL CORPORATIONS

## PATENT, TRADEMARK AND COPYRIGHT CAUSES

501 WEST BROADWAY

SUITE 1400

SAN DIEGO, CALIFORNIA 92101-3505

(619) 235-8550

FAX (619) 235-0176

INTERNET WWW.KNOB.COM

LOUIS J. KNOBBE\*  
DON W. MARTENS\*  
GORDON H. OLSON\*  
JAMES B. BEAR  
DARRELL L. OLSON\*  
WILLIAM B. BUNKER  
WILLIAM H. NIEMAN  
LOWELL ANDERSON  
ARTHUR S. ROSE  
JAMES F. LESNIAK  
NED A. ISRAELSEN  
DREW S. HAMILTON  
JERRY T. SEWELL  
JOHN B. SGANGA, JR  
EDWARD A. SCHLATTER  
GERARD VON HOFFMANN  
JOSEPH R. RE  
CATHERINE J. HOLLAND  
JOHN M. CARSON  
KAREN VOGEL WEIL  
ANDREW H. SIMPSON  
JEFFREY L. VAN HOOSEAR  
DANIEL E. ALTMAN  
ERNEST A. BEUTLER  
MARGUERITE L. GUNN  
STEPHEN C. JENSEN  
VITO A. CANUSO III  
WILLIAM H. SHREVE  
LYNDA J. ZADRA-SYMES†

STEVEN J. NATAUPSKY  
PAUL A. STEWART  
JOSEPH F. JENNINGS  
CRAIG S. SUMMERS  
ANNEMARIE KAISER  
BRENTON R. BABCOCK  
THOMAS F. SMERAL, JR  
MICHAEL H. TRENHOLM  
DIANE M. REED  
JONATHAN A. BARNEY  
RONALD J. SCHOENBAUM  
JOHN R. KING  
FREDERICK S. BERRETTA  
NANCY WAYS VENSKE  
JOHN P. GIEZENTANNER  
ADEEL S. AKHTAR  
GINGER R. DREGER  
THOMAS R. ARNO  
DAVID N. WEISS  
DANIEL HART, PH.D.  
DOUGLAS G. MUEHLHAUSER  
LORI LEE YAMATO  
STEPHEN M. LOBBIN  
ROBERT F. GAZDZINSKI  
STACEY R. HALPERN  
MICHAEL K. FRIEDLAND  
DALE C. HUNT, PH.D.  
LEE W. HENDERSON, PH.D.  
DEBORAH S. SHEPHERD

RICHARD E. CAMPBELL  
MARK M. ABUMERI  
JON W. GURKA  
KATHERINE W. WHITE  
ERIC M. NELSON  
ALEXANDER C. CHEN  
MARK R. BENEDICT, PH.D.  
PAUL N. CONOVER  
ROBERT J. ROBY  
SABING H. LEE  
KAROLINE A. DELANEY  
JOHN W. HOLCOMB  
JAMES J. MULLEN, III, PH.D.  
JOSEPH S. CIANFRANI  
JOSEPH M. REISMAN, PH.D.  
WILLIAM R. ZIMMERMAN  
GLEN L. NUTTALL  
ERIC S. FURMAN, PH.D.  
DO TE KIM  
TIRZAH ABE LOWE  
GEOFFREY Y. IIDA  
ALEXANDER S. FRANCO  
SANJIVPAL S. GILL  
SUSAN M. MOSS  
GUY PERRY  
JAMES W. HILL, M.D.  
ROSE M. THIESSEN, PH.D.  
MICHAEL L. FULLER  
MICHAEL A. GIULIANA

OF COUNSEL  
JERRY R. SEILER  
JAPANESE PATENT ATTY  
KATSUHIRO ARAI\*\*  
EUROPEAN PATENT ATTY  
MARTIN HELLEBRANDT  
KOREAN PATENT ATTY  
MINCHEOL KIM  
SCIENTISTS & ENGINEERS  
(NON-LAWYERS)  
RAIMOND J. SALENIEKS\*\*  
NEIL S. BARTFELD, PH.D. \*\*  
DANIEL E. JOHNSON, PH.D. \*\*  
JEFFERY KOEPKE, PH.D.  
KHURRAM RAHMAN, PH.D.  
JENNIFER A. HAYNES, PH.D.  
BRENDAN P. O'NEILL, PH.D.  
THOMAS Y. NAGATA  
ALAN C. GORDON  
PABLO S. HUERTA  
LINDA H. LIU  
MICHAEL J. HOLIHAN  
YASHWANT VAISHNAV, PH.D.  
MEGUMI TANAKA

Assistant Commissioner for Patents  
Washington, D.C. 20231

**CERTIFICATE OF MAILING BY "EXPRESS MAIL"**

**Attorney Docket No. :** REALNET.066A

**Applicant(s) :** Sujal Patel and Brad Hefta-Gaub

**For :** SYSTEM AND METHOD OF  
TRANSMITTING DATA PACKETS

**Attorney :** Eric M. Nelson

**"Express Mail"**

**Mailing Label No. :** EL 531 000 596 US

**Date of Deposit :** December 3, 1999

I hereby certify that the accompanying

Transmittal in Duplicate; Specification in 21 pages; four (4) sheets of drawings;  
Check for Filing Fee; Return Prepaid Postcard

are being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and are addressed to the Assistant Commissioner for Patents, Washington, D.C. 20231.

  
Eric M. Nelson

S:\DOCS\EMN\EMN-4063.DOC  
120399

**SYSTEM AND METHOD OF TRANSMITTING DATA PACKETS**Background of the Invention5     Field of the Invention

The field of the invention relates to transmitting data via a network. More particularly the invention relates to altering the transmission characteristics of a streaming media server based upon the occurrence of selected conditions.

10     Description of the Related Art

Streaming media is a method of making audio, video, and other multimedia data available in real-time, over a network. Figure 1 illustrates a typical streaming media system. If the source of the streaming media is a live performance, the performance is often recorded as a data signal by an input device, such as, for example, a microphone 102 or a camera 104. The input device then sends the data signal to an encoding system 106 which converts the data signal into a digital form and compresses the digital signal into streamable data objects 107. The streamable data objects 107 may be sent to a content creation station 108 for editing or may alternatively be sent directly to a streaming media server 110. The content creation station 108 includes content creation software, such as video editing software, that allows the user to modify the streamable data objects 107. The content creation station 108 can also be used to independently create a streamable data objects 107. After being modified or created by the content creation station 108, the streamable data objects 107 may then be transmitted to the streaming media server 110.

Once the streamable data objects 107 are located on the streaming media server 110, the streamable data objects 107 are made available for further distribution and presentation over a network 116, such as the Internet, to a number of client computers 115. One problem, however, related to transmitting streamable data objects over a network is that when the streamable media server 110 may become "overloaded" number of client computers requesting the streamable data objects 107 becomes too high. In that case, the streamable media server is not able to transmit the data packets of

the streamable data objects 107 in a timely manner. During peak hours, thousands of client computers 115 can simultaneously request the transmission of selected ones of the streamable data objects 107. In such cases, numerous of the client computers 115 may have to wait in the middle of the presentation of the respective streamable data objects for the transmission of additional data packets, thereby creating an unpleasant interruption for the user.

Therefore, there is a need for a system and method that improves the presentation of streamable data objects to users. The system should determine the system conditions and modify the process of transmitting the data packets from the streaming media server 110 to the client computers 115, thereby improving the presentation of streamable data objects to users at the client computers 115.

#### Summary of the Invention

The system and method of the present invention have several features, no single one of which is solely responsible for its desirable attributes. Without limiting the scope of this invention as expressed by the claims that follow, its more prominent features will now be discussed briefly. After considering this discussion, and particularly after reading the section entitled "Detailed Description of the Invention" one will understand how the features of this invention provide a number of advantages over traditional streaming systems.

**[TO BE COMPLETED UPON FINALIZATION OF THE CLAIMS]**

#### Brief Description of the Drawings

Figure 1 is a high level block diagram illustrating an exemplary network configuration for streaming data objects from a streaming media server to one or more client computers.

Figure 2 is a flowchart illustrating a process of modifying a transmission process based upon one or more determined system conditions.

Figure 3 is a flowchart illustrating a process of aggregating data packets in the streaming media server shown in Figure 1.

Figure 4 is a flowchart illustrating a process for determining the load of the server shown in Figure 1.

### Detailed Description of the Invention

5           The following detailed description is directed to certain specific embodiments of the invention. However, the invention can be embodied in a multitude of different ways as defined and covered by the claims.

### **System Overview**

10           Referring to Figure 1, an exemplary network configuration 100 will be described. A user communicates with a computing environment which may include multiple streaming media servers (not shown) or a single streaming media server 110 in a client/server relationship on a computer network 116. In a client/server environment, the streaming media server 110 includes a server program 112 which communicates  
15           with the client computers 115.

          The streaming media server 110, and the client computers 115 may each have one or more conventional general purpose single- or multi-chip microprocessors such as a Pentium® processor, a Pentium® Pro processor, a 8051 processor, a MIPS® processor, a Power PC® processor, or an ALPHA® processor. In addition, the microprocessor may  
20           be any conventional special purpose microprocessor such as a digital signal processor or a graphics processor. Furthermore, the streaming media server 110 and the client computers 115 may be desktop, server, portable, hand-held, set-top, or any other desired type of configuration. Furthermore, the streaming media server 110 and the client computers 115 each may be used in connection with various operating systems such as:  
25           UNIX, LINUX, Disk Operating System (DOS), OS/2, Windows 3.X, Windows 95, Windows 98, and Windows NT.

          The streaming media server, 110, and the client computers 115 may each include a network terminal equipped with a video display, keyboard and pointing device. In one embodiment of network configuration 100, one or more of the client computers 115  
30           includes a media player that is used to access the streaming media server 110.

For example, a user of a client computer 115 may utilize the media player to remotely access the server program 112 using a keyboard and/or pointing device and a visual display, such as a monitor. It is noted that although only three client computers 115 are shown in Figure 1, the network configuration 100 can include hundreds of thousands of client computers and upwards. Furthermore, it is noted that although one streaming media server 110 is shown in Figure 1, the network configuration 100 can include hundreds of thousands of streaming media servers and upwards.

The network 116 may include any type of electronically connected group of computers including, for instance, the following networks: a virtual private network, a public Internet, a private Internet, a secure Internet, a private network, a public network, a value-added network, an intranet, and the like. In addition, the connectivity to the network may be, for example, remote modem, Ethernet (IEEE 802.3), Token Ring (IEEE 802.5), Fiber Distributed Datalink Interface (FDDI) or Asynchronous Transfer Mode (ATM). The network 116 may connect to the client computer 115, for example, by use of a modem or by use of a network interface card that resides in the client computers 115.

Devices, other than the hardware configurations described above, may be used to communicate with the streaming media server 110. For example, such devices can include: a portable personal computer with a modem or wireless connection interface, a cable interface device connected to a visual display, or a satellite dish connected to a satellite receiver and a television. For convenience of description, each of the above hardware configurations are included within the definition of the client computers 115. Other ways of allowing communication between the user 102 and the streaming media server 110 are envisioned.

Further, it is noted the streaming media server 110 and the client computers 115, may not necessarily be located in the same room, building or complex. In fact, the streaming media server 110 and the client computers 115 could each be located in different states or countries.

The server program 112 comprises various modules, each of the modules comprising various sub-routines, procedures, definitional statements, and macros. Each of the modules are typically separately compiled and linked into a single executable program.

Thus, the processes that are undergone by selected ones of the modules may be arbitrarily redistributed to one of the other modules, combined together in a single module, or made available in a shareable dynamic link library.

The server program 112 transmits selected streamable data objects 107 to selected client computers 115. The streamable data objects 107 can be formatted according to any streaming data format and can include all or part of any type of media content such as: a movie, a picture, a slide show, music, a broadcasting program, headline information, or stock quotes. Upon a request from a selected one of the client computers 115, the server program 112 transmits one or more data packets that collectively define the streamable data objects to the selected client computer so that the media player for the selected client computer can render the streamable data object to a user at the client computer. The server program 112 can transmit the data packets using one of several network protocols including: UDP, IP, TCP, ATM, or frame relay.

The server program 112 may be written in any programming language such as C, C++, BASIC, Pascal, Java, and FORTRAN and ran under any of a wide variety of operating systems. C, C++, BASIC, Pascal, Java, and FORTRAN are industry standard programming languages for which many commercial compilers can be used to create executable code.

### **Method of Operation**

Figure 2 is a high level flowchart illustrating a process for modifying a streaming media process based upon selected system conditions. Before starting the step 200, the streaming media server 110 has received one or more streamable media objects 107. Furthermore, selected ones of the client computers 115 have requested the streaming media server 110 to transmit or "stream" one or more of the streamable data objects 107 to the client computers 115. It is noted that depending on the embodiment, selected steps can be omitted from Figure 2 and others can be added.

After starting at a step 200, the streaming media server 110 (Figure 1) proceeds to a step 204. At the step 204, the streaming media server 110 processes one or more network events. In one embodiment of the invention, the network events can include receiving a command to perform one of the following actions: play one of the streamable media objects 107, pause a presentation one of the streamable media objects

107, seek to a selected location in one of the streamable media objects 107, or re-send a selected data packet. Furthermore, the network events can include receiving a “ping” message from one of the client computers 115. It is noted that a ping message may be used by the client computers 115 to indicate to the streaming media server 110 that the client computers 115 are still operational.

Next, at a step 208, the streaming media server 110 determines one or more system conditions of the streaming media server 110. In one embodiment of the invention, the occurrence of selected system conditions identifies for the streaming media server 110 a need to adjust the process of transmitting the streamable media data objects 107 to the client computers 115. For example, the determined system conditions can include: (i) the number of forced processings of network events (described in further detail below with respect to decision step 228), (ii) the number of client computers 115 that are behind their scheduled delivery time, (iii) the number of client computers 115 that have requested streamable data objects 107, (iv) the total byte count of the streamable data objects 107 that have been requested by the client computers 115, (v) the number of the streamable data objects 107 that have been requested by the client computers 115, (vi) the number of streamable data objects 107 that are maintained by the streaming media server 110, and/or (vii) the actual transmission rate of the streaming media server 110 with respect to the client computers 115.

If the streaming media server 110 (Figure 1) determines to modify the transmission process of the streamable data objects 107, the streaming media server 110 proceeds to a step 216. At the step 216, the streaming media server 110 modifies one or more of the transmission characteristics of the streaming media server 110. Examples of adjustable transmission characteristics can include: the aggregation or non-aggregation of one or more smaller sized data packets into a larger sized data packet; increasing or decreasing the frequency of transmission of the data packets; increasing or decreasing the size of the data packets when the data packets are initially prepared; changing the network operating system call that is used to transmit and/or receive the data packets, for example, from asynchronous to synchronous or vice-versa; increasing or decreasing the number of channels in the streaming media server 110; and/or changing the types of channels in the streaming media server 110. In one embodiment



of the invention, a channel is a network socket. Different types of channels can include various sockets under TCP, IP, or UDP.

Referring again to the decision step 212, if the streaming media server 110 determines not to modify the transmission process based upon the determined system conditions, or, alternatively, after completion of step 216, the streaming media server 110 proceeds to a step 220 wherein the streaming media server 110 transmits selected data packets to the client computers 115. In one embodiment of the invention, the streaming media server 110 transmits selected packets according to a priority scheme. In this embodiment, each of the data packets has an associated transmission deadline time. The transmission deadline time that is associated with the data packet indicates a point in time whereby the streaming media server 110 should have transmitted the data packet to a respective one of the client computers 115. In one embodiment, the streaming media server 110 transmits each of the data packets having the highest priority according to the transmission deadline times that are respectively associated with the data packets.

Next, the streaming media server 110 proceeds to a decision step 224. At the decision step 224, the streaming media server 110 determines whether it has transmitted each of the data packets that are due according to the transmission deadline times that are respectively associated with each of the data packets. If the streaming media server 110 determines that it has processed all of the data packets that are due, the streaming media server 110 returns to the step 204 to process any new network events that have occurred.

Referring again to the decision step 224, if the streaming media server 110 determines that all of the data packets that are due have not been processed, the streaming media server 110 proceeds to a decision step 228.

At the decision step 228, the streaming media server 110 determines whether a time out threshold has been exceeded. It is noted that in periods of high load for the streaming media server 110, the streaming media server 110 will not be able to transmit all of the streamable data objects 107 to the client computers 115 in a timely manner. To prevent the streaming media server 110 from occupying all of its time transmitting data, the streaming media server 110 employs a watch dog timer that informs the

streaming media server 110 when the streaming media server 110 has been processing and transmitting outgoing data packets for a selected time interval. In one embodiment of the invention, the selected time interval is about 1/3 of a second. In another embodiment of the invention, the selected time interval is between about 1/4 of a second and about 3/4 of a second. In yet another embodiment of the invention, the selected time interval is between about 1/5 of a second and about 7/8 of a second.

Still referring to the decision step 228, if the streaming media server 110 determines that the time out threshold has not been exceeded, the streaming media server 110 returns to the step 220 to process additional data packets. However, if the streaming media server 110 determines that the time out value threshold has been exceeded, the streaming media server 110 interrupts the transmission of the data packets that are due and returns to the step 204. It is noted that an interruption of the transmission of the data packets to process the network events is herein referred to as a “forced processing of a network event.”

Figure 3 is a flowchart illustrating one embodiment of a process of determining the server load. Figure 3 illustrates in further detail the steps that occur in step 208 of Figure 2 for one embodiment of the invention. In this embodiment, the streaming media server 110 maintains a state machine to describe the current system load. The state machine comprises three states to describe the system load, namely, NORMAL, HIGH, and EXTREMELY HIGH. It is to be appreciated that fewer or more states may be used. Furthermore, it is to be appreciated that other descriptions for the states may be used, *e.g.*, LOW, VERY LOW, ABOVE NORMAL, DANGEROUS. It is noted that depending on the embodiment, selected steps can be omitted from Figure 3 and others can be added.

After starting at a step 300, the streaming media server 110 proceeds to a step 304 wherein the streaming media server 110 determines the processing percentage of the forced processings of network events that occur within the streaming media server 110. The processing percentage is defined as the number of the forced processings of network events that occur within the streaming media server 110 divided by the total number of network events. It is noted that as the percentage of forced processings gets higher, this indicates that the streaming media server 110 is spending more of its time in

transmitting data packets as opposed to other activities, such as processing network events. Accordingly, if the percentage of forced processings exceeds a processing threshold, the streaming media server 110 assumes that the server load is HIGH.

Moving to a step 308, the streaming media server 110 determines whether the percentage of forced processings of network events exceeds the processing threshold. It is noted that, in one embodiment of the invention, the processing threshold is about 70%. In another embodiment of the invention, the processing threshold is between about 65% and about 75%. In yet another embodiment of the invention, the processing threshold is between about 55% and about 95%.

If the streaming media server 110 determines that the percentage of forced processings of network events exceeds the processing threshold, the streaming media server 110 determines that the load of the server is either HIGH or EXTREMELY HIGH. In this event, the streaming media server 110 then proceeds to a decision step 312 to more particularly determine the server load. It is noted that in one embodiment, the state of the load of the streaming media server 110 is initially set to NORMAL.

At the decision step 312, the streaming media server 110 determines whether the percentage of client computers 115 that are behind in their scheduled delivery time is greater than a scheduling threshold. The scheduling threshold is defined as the number of client computers 115 that are behind in their scheduled delivery time divided by the total number of client computers 115 that are being serviced. If the streaming media server 110 determines that the percentage of client computers 115 that are behind in their scheduled delivery time is greater than a scheduling percentage threshold, the streaming media server 110 proceeds to a step 318, and the streaming media server 110 determines that the state of the load of the streaming media server 110 is EXTREMELY HIGH. The streaming media server 110 then moves to an end step 320, and the process flow continues at the decision step 212 of Figure 2.

Referring again to the decision step 312 of Figure 3, if the streaming media server 110 determines that the percentage of client computers 115 that are behind in their scheduled delivery time is less than or equal to the scheduling percentage threshold, the streaming media server 110 proceeds to a step 316, and the streaming

media server 110 determines that the state of the server load is HIGH. The process flow then continues to the end step 320 (discussed in further detail above).

Referring again to the decision step 308, if the streaming media server 110 (Figure 1) determines that the percentage of forced processings of network events is less than or equal to the processing threshold, the streaming media server 110 proceeds to a decision step 324. At the decision step 324, the streaming media server 110 determines whether the streaming media server 110 had been in a overload state at any point during a previous predefined duration. In one embodiment of the invention, the predefined duration is about 5 minutes. In another embodiment of the invention, the predefined duration is about between 4 and 1/2 minutes and about 5 and 1/2 minutes. In yet another embodiment of the invention, the predefined duration is about between 1 and 8 minutes. Furthermore, in one embodiment of the invention, the streaming media server 110 defines an overload state to include the states HIGH and EXTREMELY HIGH.

In the embodiment shown with respect to step 324, the streaming media server 110 maintains an overload state until the streaming media server 110 has experienced a below-threshold percentage of forced processings for the predefined duration.

If the streaming media server 110 detects an overload state for the previous predefined duration, the streaming media server 110 determines that sufficient time has not yet passed to downgrade the status of the server load, and the server 110 proceeds to the end step 320 (discussed further above). However, if the streaming media server 110 was not in an overload state for the previous predefined duration, the streaming media server 110 proceeds to a step 328. At the step 328, the streaming media server 110 reduces the severity of the state of the server load. For example, if the state of the server load is EXTREMELY HIGH, the streaming media server 110 can reduce the state of the server load to HIGH. Furthermore, for example, if the state of the server load is HIGH, the streaming media server 110 can reduce the load to NORMAL. The streaming media server 110 then proceeds to the end step 320 (discussed in further detail above).

Figure 4 is a flowchart illustrating a process for aggregating packets. By aggregating packets into an aggregated packet, the streaming media server 110 improves the throughput of the data packets to the client computers 115, thereby

improving quality of the presentation of the streamable data objects 107 by the client computers 115. Aggregation of data packets ultimately reduces the number of data packets per second, therefore reducing the number of operating system calls and the overall system/CPU load. It is noted that although packet aggregation improves the throughput of data packets to the client computers 115, selected ones of the client computers 115 may encounter a reduction in the quality of the presentation if one or more of the aggregated data packets are “lost” in transmission.

Figure 4 describes in further detail the steps that occur within step 220 of Figure 2. It is noted that depending on the embodiment, selected steps can be omitted from Figure 4 and others can be added. After starting at a step 400, the streaming media server 110 (Figure 1) proceeds to a step 412, wherein the streaming media server 110 determines whether the system load exceeds a selected threshold. In one embodiment of the invention, if the state of the load is either HIGH or EXTREMELY HIGH, the streaming media server 110 determines that the load is above the selected threshold. Furthermore, if the streaming media server 110 determines that the server load is below the selected threshold.

If the streaming media server 110 (Figure 1) determines that the load does not exceed the selected threshold, the streaming media server 110 proceeds to a step 416. At the step 416, the streaming media server 110 transmits a packet of the streaming media server 110 to one of the client computers 115. The streaming media server 110 then proceeds to an end step 436, and process flow returns the decision step 224 of Figure 2.

Referring again to the decision step 412, if the streaming media server 110 determines that the server load exceeds the selected threshold, the streaming media server proceeds to a step 420.

At the step 420, the streaming media server 110 aggregates one or more data packets into an aggregated data packet. In one embodiment of the invention, the streaming media server 110 selects each of the data packets that are due for transmission according to a requested transmission time that is associated with each of the packets.

Proceeding to a decision step 424, the streaming media server 110 determines whether the size of the aggregated data packet is greater than an aggregation threshold.

The size of the aggregation threshold can either be predefined or alternatively, dynamically determined. In one embodiment of the invention, the size of the aggregation threshold is based upon the state of the load of the streaming media server 110 as well as the actual delivery rate between the streaming media server 110 and the client computer 115. In one embodiment of the invention, the sizes of the aggregation thresholds are set forth below in Table 1.

**Table 1**

Conditions	Minimum Threshold	Maximum Threshold
server load = NORMAL	200	300
server load = HIGH and actual delivery rate < 100,000	700	1000
(Server load = HIGH and actual deliver rate >= 1000,000) or server load = EXTREMELY HIGH	1000	1350

As is shown in Table 1, depending on selected conditions, the streaming media server 110 aggregates data packets until the size of the aggregated data packet exceeds the minimum threshold without exceeding the maximum threshold. It is noted that if the size of the aggregated packet is below the minimum threshold, but another packet cannot be added without exceeding the maximum threshold, the streaming media server 110 will not add the data packet to the aggregated data packet. Furthermore it is noted that the actual delivery rate is defined as the average of the total number of kilobytes per second being delivered by the streaming media server 110 to the client computers 115.

In one embodiment of the invention, the streaming media server 110 does not generate an aggregated data packet whose size is greater than the maximum transmission unit (MTU) for the network configuration 100 (Figure 1). The MTU is defined as the largest packet is that is transported by intermediary network devices. If a data packet is larger than the MTU, the intermediary network devices partitions the data packet into multiple data packets, each of an acceptable size. However, as can be appreciated by one of ordinary skill in the art, if a partitioned data packet is lost, all of the other partitioned data packets are unusable by the client computer 115, even if

received. Accordingly, in this embodiment, the invention advantageously avoids that problem.

Still referring to the decision step 424, if the streaming media server 110 determines that the size of the aggregated data packet exceeds the selected threshold, the streaming media server 110 proceeds to a step 432. At the step 432, the streaming media server 110 transmits the aggregated data packet to respective ones of the client computers 115. The streaming media server 110 then proceeds to the end step 436, and process flow returns to the decision step 224 of Figure 2.

Referring again to the decision step 424 of Figure 4, if the streaming media server 110 determines that size of the aggregated packet is less than the threshold, the streaming media server 110 returns to the step 420 to add another data packet to the aggregated data packet.

Advantageously, the present invention allows for the modification of a process of transmitting data packets based upon the system conditions of the streaming media server. For example, during periods of high or extreme load, the streaming media server 110 can aggregate data packets into an aggregated data packets. It is noted that the aggregating data packets into an aggregated data packet may reduce the quality of presentation for selected user, data packet aggregation decreases the overall server load and increases the quality of the presentation of the streaming data objects to a majority of the client computers 115.

WHAT IS CLAIMED IS:

1. A method of transmitting a plurality of data packets from a server computer to at least one client computer, the method comprising:

5 determining one or more system conditions of the server computer; and  
modifying a process of transmitting the data packets from the server computer to the client computer, the modifying based at least in part upon the determined system conditions.

10 2. The method of Claim 1, wherein determining one or more system conditions comprises determining a server load that is associated with the server computer.

15 3. The method of Claim 2, wherein determining the server load comprises comparing the number of data packets that are overdue to the total number of data packets.

20 4. The method of Claim 2, wherein determining the server load comprises comparing the number of network events processed by a server program that is executing on the server computer due to exceeding a time out threshold to the total number of network events that the server program processes.

25 5. The method of Claim 4, wherein the network events are selected from the group comprising: a play command, a pause command, a seek command, a ping command, and a re-send command.

6. The method of Claim 2, wherein the server load is based at least in part upon the actual transmission rate between the server computer and the client computer.

30 7. The method of Claim 1, wherein modifying the process of transmitting the streamable data objects from the server computer to the client computer comprises aggregating one or more data packets into an aggregated data packet.



8. The method of Claim 7, wherein the data packets are not aggregated larger than the size of a maximum transmission unit for any intermediary network device that is in the transmission path between the server computer and the client computer.

9. The method of Claim 7, wherein the data packets are aggregated in an aggregated data packet until the size of the aggregated data packet exceeds a minimum threshold without exceeding a maximum threshold.

10. The method of Claim 9, wherein the size of the minimum threshold or the maximum threshold is dependent on the load of the server computer.

11. The method of Claim 10, wherein the minimum threshold is about 200 bytes and wherein the maximum threshold is about 300 bytes.

12. The method of Claim 10, wherein the minimum threshold is about 700 bytes and wherein the maximum threshold is about 1000 bytes.

13. The method of Claim 10, wherein the minimum threshold is about 1000 bytes and wherein the maximum threshold is about 1350 bytes.

14. The method of Claim 1, wherein modifying the process of transmitting the streamable data objects from the server computer to the client computer comprises increasing the packet size of one or more data packets that are used to transmit the streamable data objects.

15. The method of Claim 1, wherein modifying the process of transmitting the streamable data objects from the server computer to the client computer comprises increasing or decreasing the number of channels that are used to transmit the streamable data objects.

16. The method of Claim 1, wherein modifying the process of transmitting the streamable data objects from the server computer to the client computer comprises

either increasing or decreasing the frequency of transmission of one or more data packets that are used to transmit the streamable data objects.

5           17.    A server computer for transmitting data packets, the server computer comprising:

          a plurality of data packets; and

          a server program for determining one or more system conditions and for modifying a process of transmitting the data packets from the server computer to a client computer, the modifying based at least in part upon the determined  
10           system conditions.

          18.    The system of Claim 17, wherein the data packets collectively comprise a streamable data object.

15           19.    The system of Claim 17, wherein determining one or more of the system conditions comprises determining a server load that is associated with the server computer.

20           20.    The system of Claim 19, wherein determining the server load comprises comparing the number of data packets that are overdue to the total number of data packets.

25           21.    The system of Claim 19, wherein determining the server load comprises comparing the number of network events processed by a server program that is executing on the server computer due to exceeding a time out threshold to the total number of network events that the server program processes.

30           22.    The system of Claim 17, wherein the one or more network events is selected from the group comprising: a play command, a pause command, a seek command, a ping command, and a re-send command.

23. The system of Claim 17, wherein modifying the process of transmitting the streamable data objects from the server computer to the client computer comprises increasing the packet size of one or more data packets.

5           24. The system of Claim 17, wherein modifying the process of transmitting the streamable data objects from the server computer to the client computer comprises increasing the number of channels that are used to transmit the streamable data objects.

10           25. The system of Claim 17, wherein modifying the process of transmitting the streamable data objects from the server computer to the client computer comprises either increasing or decreasing the frequency of transmission of one or more data packets.

15           26. A system for transmitting data packets from a server computer to at least one client computer, the system comprising:

          means for determining one or more system conditions; and

          means for modifying a process of transmitting the data packets from the server computer to the client computer, the modifying based at least in part upon the determined system conditions.

20           27. The system of Claim 26, wherein the means of determining one or more system conditions comprises means for determining a server load that is associated with the server computer.

25           28. A system for aggregating data packets, the system comprising:  
          a plurality of data packets that collectively comprise one or more streamable data objects; and

          a server computer operably connected to a client computer via a network, the server computer transmitting the data objects to the server computer, the server computer periodically determining, based upon the load of the server computer, whether to aggregate one or more of the data packets into an aggregated data packet.

30

29. The system of Claim 28, wherein determining the server load comprises comparing the number of data packets that are overdue to the total number of data packets

5

30. The system of Claim 28, wherein determining the server load comprises comparing the number of network events processed by a server program that is executing on the server computer due to exceeding a time out threshold to the total number of network events that the server program processes.

10

31. The system of Claim 30, wherein the network events are selected from the group comprising: a play command, a pause command, a seek command, a ping command, and a re-send command.

15

32. The system of Claim 28, wherein the server load is based at least in part upon the actual transmission rate between the server computer and the client computer.

20

33. The method of Claim 28, wherein the data packets are not aggregated larger than the size of a maximum transmission unit for any intermediary network device that is in the transmission path between the server computer and the client computer.

25

34. The method of Claim 28, wherein the data packets are aggregated in an aggregated data packet until the size of the aggregated data packet exceeds a minimum threshold without exceeding a maximum threshold.

30

35. The method of Claim 34, wherein the size of the minimum threshold relates to a quality of presentation of the streamable data objects and the maximum threshold relates to a maximum transmission unit.

36. The method of Claim 34, wherein the size of the minimum threshold or the maximum threshold is dependent on the load of the server computer.

37. The system of Claim 36, wherein the minimum threshold is about 200 bytes and wherein the maximum threshold is about 300 bytes.

38. The system of Claim 36, wherein the minimum threshold is about 700 bytes and wherein the maximum threshold is about 1000 bytes.

39. The system of Claim 36, wherein the minimum threshold is about 1000 bytes and wherein the maximum threshold is about 1350 bytes.

40. A method of aggregating data packets, the method comprising:  
determining, based upon the load of a server computer, whether to  
aggregate one or more of the data packets into an aggregated data packet; and  
transmitting the aggregated data packet to a client computer.

41. The method of Claim 40, wherein the data packets are not aggregated in an aggregated data packet larger than the size of a maximum transmission unit for any intermediary network device that is in the transmission path between the server computer and the client computer.

42. The method of Claim 40, wherein determining the server load comprises comparing the number of data packets that are overdue to the total number of data packets.

43. The method of Claim 40, wherein determining the server load comprises comparing the number of network events processed by a server program that is executing on the server computer due to exceeding a time out threshold to the total number of network events that the server program processes.

44. The method of Claim 40, wherein the network events are selected from the group comprising: a play command, a pause command, a seek command, a ping command, and a re-send command.

45. The method of Claim 40, wherein the server load is based at least in part upon the actual transmission rate between the server computer and the client computer.

5 46. The method of Claim 40, wherein the data packets are aggregated in an aggregated data packet until the size of the aggregated data packet exceeds a minimum threshold without exceeding a maximum threshold.

10 47. The method of Claim 46, wherein the size of the minimum threshold or the maximum threshold is dependent on the load of the server computer.

48. The method of Claim 47, wherein the minimum threshold is about 200 bytes and wherein the maximum threshold is about 300 bytes.

15 49. The method of Claim 47, wherein the minimum threshold is about 700 bytes and wherein the maximum threshold is about 1000 bytes.

50. The method of Claim 47, wherein the minimum threshold is about 1000 bytes and wherein the maximum threshold is about 1350 bytes.

20 51. The method of Claim 46, wherein the size of the minimum threshold relates to a quality of presentation of the streamable data objects and the maximum threshold relates to a maximum transmission unit.

# SYSTEM AND METHOD OF TRANSMITTING DATA PACKETS

## Abstract of the Disclosure

A system and method of transmitting data packets. The system determines one or more system conditions of the server computer and modifies a process of transmitting the data packets from a server computer to a client computer, the modifying based at least in part upon the determined system conditions. The determined system conditions can include: (i) the number of forced processings of network events, (ii) the number of clients computers that are behind their scheduled delivery time, (iii) the number of client computers that have requested streamable data objects, (iv) the total byte count of the streamable data objects that have been requested by the client computers, (v) the number of the streamable data objects that have been requested by the client computers, (vi) the number of streamable data objects that are maintained by the streaming media server, and/or (vii) the actual transmission rate of the streaming media server with respect to the client computers. In one embodiment of the invention, the server computer aggregates data packets prior to transmitting the data packets to a client computer when the load of the server computer becomes high or extremely high.

PATENT

S:\DOCS\EMN\EMN-4054.DOC  
113099

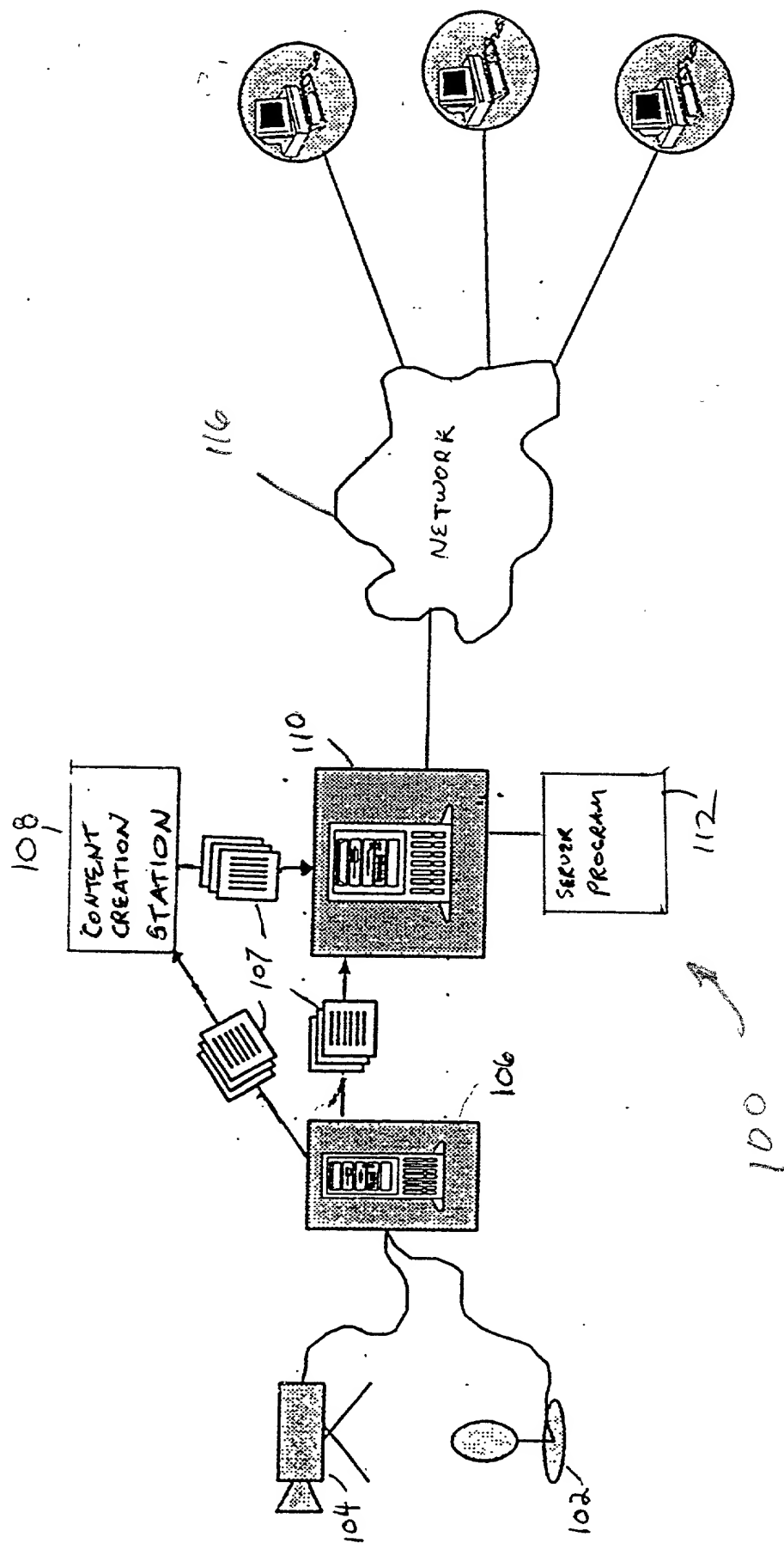
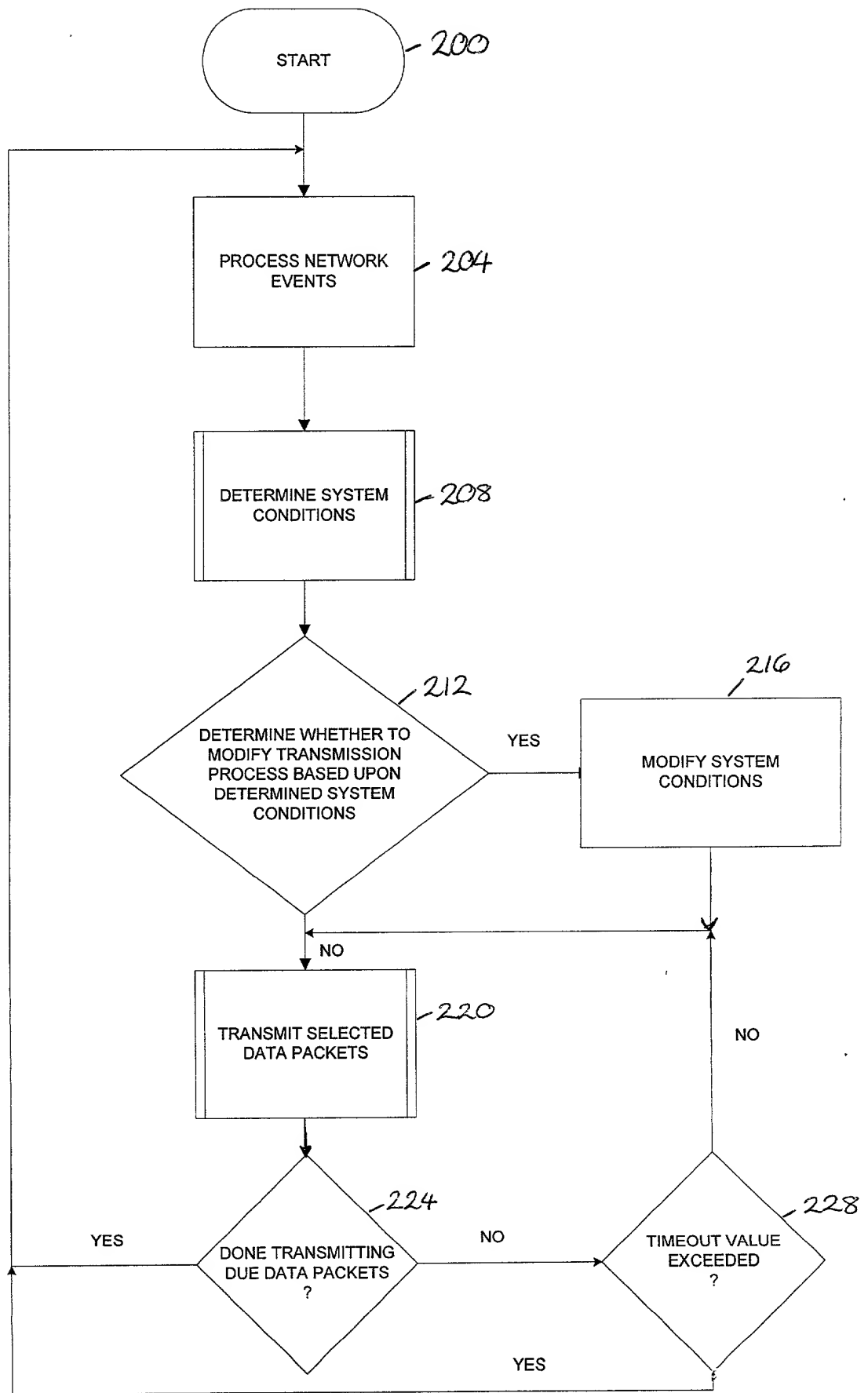


Figure 1



FIG. 2



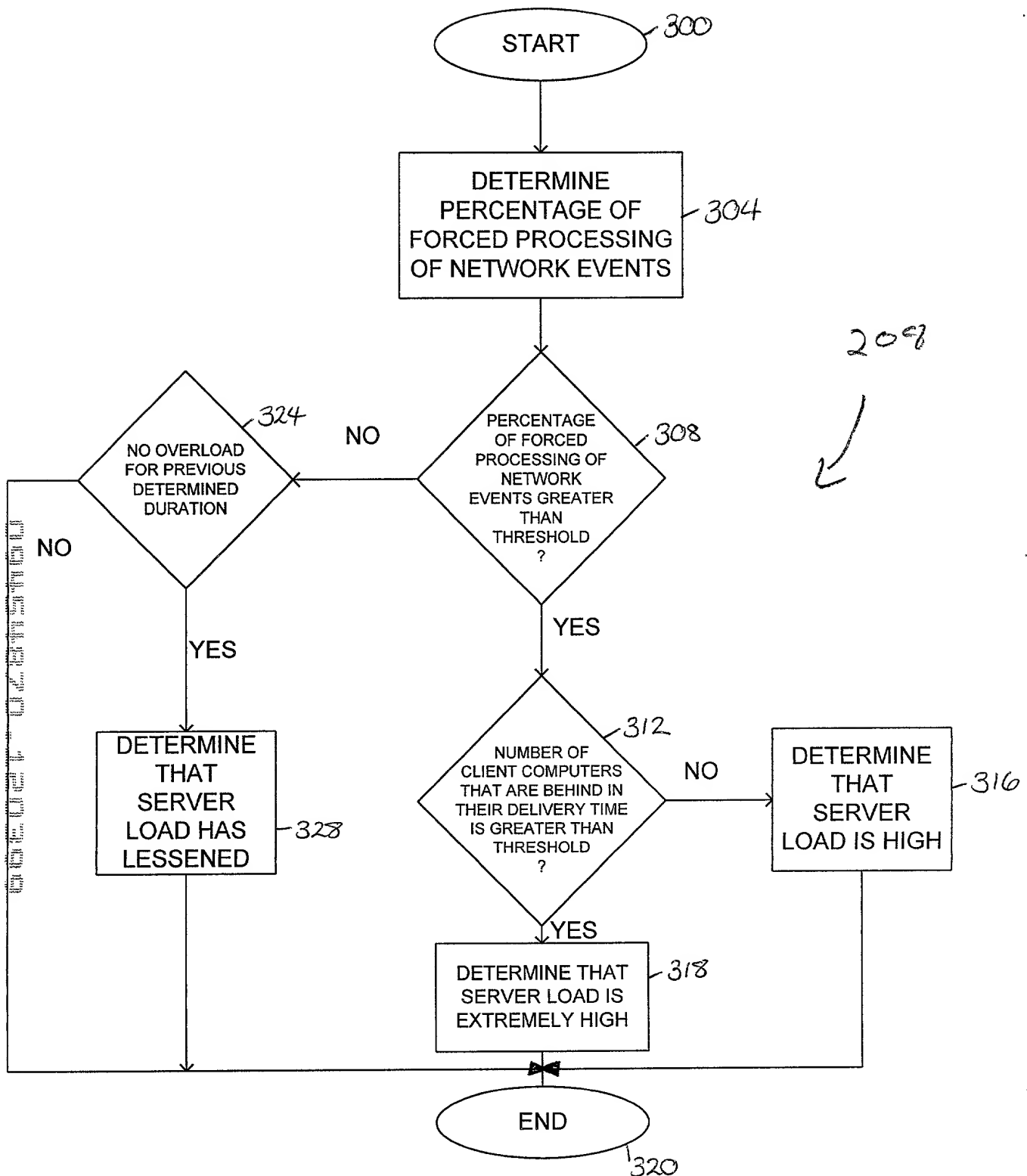


FIG. 3

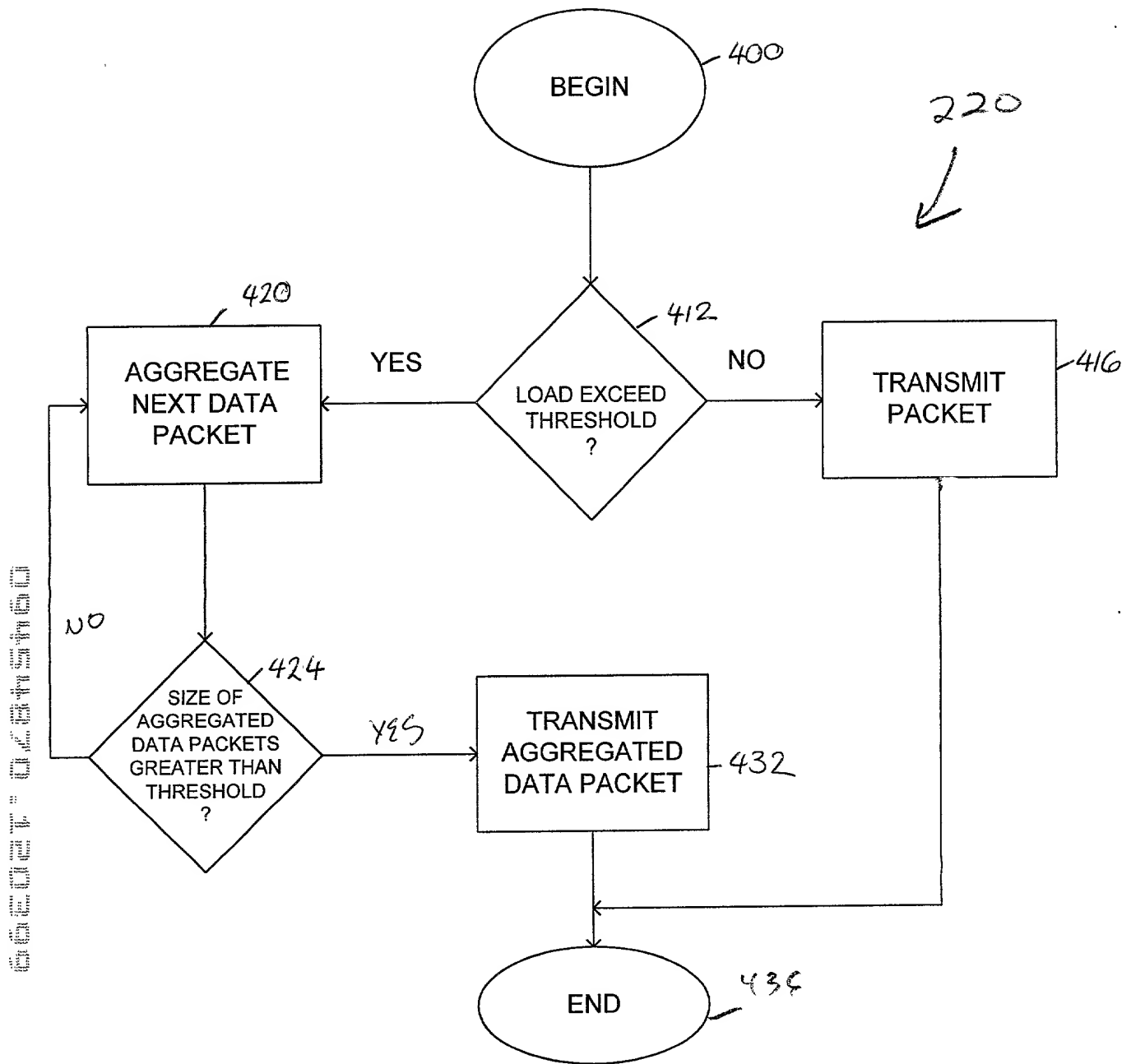


FIG. 4